

BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	SSSSSSSS
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	SSSSSSSS
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	SSSSSSSS
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		00000000		00000000		TTT	SSS	SSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTT	SSS	SSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTT	SSS	SSSSSSSS

```
BBBBBBBBB 000000 000000 TTTTTTTTTT SSSSSSSS
BBBBBBBBB 000000 000000 TTTTTTTTTT SSSSSSSS
BB      BB 00      00 00      00 TT      SS
BB      BB 00      00 00      00 TT      SS
BB      BB 00      00 00      00 TT      SS
BB      BB 00      00 00      00 TT      SS
BBBBBBBBB 00      00 00      00 TT      SSSSSS
BBBBBBBBB 00      00 00      00 TT      SSSSSS
BB      BB 00      00 00      00 TT      SS
BB      BB 00      00 00      00 TT      SS
BB      BB 00      00 00      00 TT      SS
BB      BB 00      00 00      00 TT      SS
BBBBBBBBB 000000 000000 TT      SSSSSSSS
BBBBBBBBB 000000 000000 TT      SSSSSSSS
```

```
....
....
....
....
```

```
MM      MM  AAAAAA RRRRRRRR
MM      MM  AAAAAA RRRRRRRR
MMMM  MMMM AA      AA RR      RR
MMMM  MMMM AA      AA RR      RR
MM  MM  MM  AA      AA RR      RR
MM  MM  MM  AA      AA RRRRRRRR
MM  MM  MM  AA      AA RRRRRRRR
MM  MM  MM  AAAAAAAAAA RR  RR
MM  MM  MM  AAAAAAAAAA RR  RR
MM  MM  MM  AA      AA RR      RR
MM  MM  MM  AA      AA RR      RR
MM  MM  MM  AA      AA RR      RR
MM  MM  MM  AA      AA RR      RR
```

.TITLE BOOTS MACROS
.IDENT 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

FACILITY: BOOTS

ABSTRACT:

This module contains macros for the BOOTS facility

ENVIRONMENT:

AUTHOR: STEVE BECKHARDT, CREATION DATE: 31-Oct-1979

MODIFIED BY:

V03-005	KDM0073	Kathleen D. Morse	22-Aug-1983
		Change TIMEDWAIT macro to include load address for VMB (e.g., address of RPB).	
V03-004	KDM0058	Kathleen D. Morse	13-Jul-1983
		Add boot-time specific TIMEDWAIT macro for boot drivers.	
V03-003	KTA3058	Kerbey T. Altmann	20-Jun-1983
		Add cell for boot device name (may be different from driver name!). Also unit disconnect routine.	
V03-002	KTA3034	Kerbey T. Altmann	02-Feb-1983
		Add cell for booting node name.	

.SBTTL DECLARATIONS

INCLUDE FILES:

MACROS:

\$BOOT_DRIVER MACRO - SETS UP A TABLE ENTRY FOR A BOOT DEVICE DRIVER.
EACH TABLE ENTRY CONTAINS:

CPUTYPE CPU TYPE. DEFAULT = -1 (DON'T CARE)
 DEVTYPER BOOT DEVICE TYPE VALUE. DEFAULT = -1 (DON'T CARE)
 ACTION ACTION ROUTINE ADDRESS (ACTUALLY OFFSET FROM
 START OF TABLE ENTRY). DEFAULT = 0 (NONE).
 SIZE SIZE OF ENTIRE DRIVER IN BYTES. CANNOT BE DEFAULTED.
 ADDR ADDRESS OF DRIVER (ACTUALLY OFFSET FROM START
 OF TABLE ENTRY). CANNOT BE DEFAULTED.
 ENTRY ADDRESS OF DRIVER ENTRY POINT (ACTUALLY OFFSET
 FROM ADDRESS OF DRIVER). DEFAULT = 0 (ADDRESS
 OF DRIVER AND ENTRY POINT ARE THE SAME).
 DRVRNAME ADDRESS OF DRIVER NAME IN .ASCIC. (ACTUALLY
 OFFSET FROM ADDRESS OF DRIVER). CANNOT BE
 DEFAULTED.
 AUXDRNAME AUXDRNAME ROUTINE ADDRESS (ACTUALLY OFFSET FROM
 START OF TABLE ENTRY). DEFAULT = 0 (NONE).
 UNIT_INIT UNIT_INIT ROUTINE ADDRESS (ACTUALLY OFFSET FROM
 START OF TABLE ENTRY). DEFAULT = 0 (NONE).
 UNIT_DISC UNIT_DISCONNECT ROUTINE ADDRESS (ACTUALLY OFFSET
 FROM START OF TABLE ENTRY). DEFAULT = 0 (NONE).
 DEVNAME BOOT DEVICE NAME ADDRESS (ACTUALLY OFFSET FROM
 START OF TABLE ENTRY). DEFAULT = FIRST TWO
 LETTERS OF DRVRNAME.

.MACRO \$BOOT_DRIVER CPUTYPE=-1,DEVTYPER=-1,ACTION,SIZE,ADDR,-
 ENTRY,DRVRNAME,AUXDRNAME,UNIT_INIT,-
 UNIT_DISC,DEVNAME

\$TABLE=. .PSECT BOOTDRVR_4

.WORD CPUTYPE
 .IF EQ CPUTYPE
 .ERROR 0 ; CPU TYPE CANNOT BE 0 ;
 .ENDC
 .WORD DEVTYPER

```

      .IF      B      ACTION
      .LONG    0
      .IFF
      .LONG    ACTION-STABLE
      .ENDC
      .LONG    SIZE
      .LONG    ADDR-STABLE
      .IF      B      ENTRY
      .LONG    0
      .IFF
      .LONG    ENTRY-ADDR
      .ENDC
      .LONG    DRVRNAME-ADDR
      .IF      B      AUXDRNAME
      .LONG    0
      .IFF
      .LONG    AUXDRNAME-ADDR
      .ENDC
      .IF      B      UNIT_INIT
      .LONG    0
      .IFF
      .LONG    UNIT_INIT-ADDR
      .ENDC
      .IF      B      UNIT_DISC
      .LONG    0
      .IFF
      .LONG    UNIT_DISC-ADDR
      .ENDC
      .IF      B      DEVNAME
      .LONG    DRVRNAME-ADDR+1
      .IFF
      .LONG    DEVNAME-ADDR
      .ENDC
      .PSECT   BOOTDRVR 2
      .ENDM    $BOOT_DRIVER

```

```

: Define the offsets into the argument list passed by VMB to SYSBOOT
:

```

```

      .MACRO   $VMBARGDEF,GBL

```

```

      $DEFINI  VMB,GBL,4

```

\$DEF	VMBSQ_FILECACHE	.BLKQ	1	:	FILERead Cache Descriptor
\$DEF	VMBSL_LO_PFN	.BLKL	1	:	Lowest PFN found by VMB
\$DEF	VMBSL_HI_PFN	.BLKL	1	:	Highest PFN exclusive
\$DEF	VMBSQ_PFNMAP	.BLKQ	1	:	PFN Bitmap descriptor
\$DEF	VMBSQ_UCODE	.BLKQ	1	:	Loaded ucode descriptor
\$DEF	VMBSB_SYSTEMID	.BLKB	6	:	48 bit SCS systemid
		.BLKW	1	:	Spare
\$DEF	VMBSL_FLAGS	.BLKL	1	:	Word of flags
\$DEF	VMBSL_CI_HIPFN	.BLKL	1	:	Highest PFN used by CI code
\$DEF	VMBSQ_NODENAME	.BLKQ	1	:	Booting node name
\$DEF	VMBSQ_ARGBYTCNT			:	Size of argument list in bytes
\$EQU	VMBSV_LOAD_SCS		0	:	Flag to SYSBOOT to load SCS

```
$DEFEND VMB,GBL,ARGDEF
```

```
.ENDM $VMBARGDEF
```

```
..
```

```
TIMEDWAIT - Timed Wait Loop with Imbedded Tests
```

```
Macro to wait for a specified interval of time. Uses a processor
specific value established by system bootstrap to determine an
approximate interval of time to wait instead of reading the
processor clock. Instructions that test for various exit conditions
may be imbedded within the wait loop, if so desired.
```

```
This version of TIMEDWAIT is set up to be used with boot drivers.
It contains the right kind of PIC references to EXESGL_TENUSEC and
EXESGL_UBDELAY, for code that is moved within the address space of
an image at run-time instead of remaining bound to the relative
offset within the image given it at link-time. (Note that these
two counters are kept in the BQO structure and are referenced via
BQOSL_UBDELAY and BQOSL_TENUSEC.)
```

```
INPUTS:
```

```
TIME - the number of 10 micro-second intervals to wait
INS1 - first instruction to imbed within wait loop
INS2 - second instruction to imbed within wait loop
INS3 - third instruction to imbed within wait loop
INS4 - fourth instruction to imbed within wait loop
INS5 - fifth instruction to imbed within wait loop
INS6 - sixth instruction to imbed within wait loop
DONELBL - label for exit from wait loop
IMBEDLBL - Label for imbedded instructions in wait loop
UBLBL - Label for UNIBUS SOBGTR loop
```

```
OUTPUTS:
```

```
R0 - indicates success of failure status. Success is defined as
the bit being at the specified sense within the specified
time interval.
R1 - destroyed, all other registers preserved.
```

```
--
```

```
.MACRO TIMEDWAIT TIME,INS1,INS2,INS3,INS4,INS5,INS6,DONELBL,?IMBEDLBL,?UBLBL
```

```
.nlist cnd
MOVL RPB$ IOVEC(R9),R1 ; Get address of IOVEC data cells.
MULL3 TIME,BQOSL_TENUSEC(R1),R1 ; Calculate time.
MOVZWL #SS$ NORMAC,R0 ; Assume success.
CLRL -(SP) ; Reserve space for delay loop index.
```

```
IMBEDLBL:
'INS1'
'INS2'
'INS3'
```



```

      'INS4'
      'INS5'
      'INS6'
      ADDL3  RPB$L_IOVEC(R9), -      ; Get address of IOVEC data cells
      MOVL   #BQO$C_UBDELAY,(SP)    ; holding delay loop cnt.
      UBLBL: SOBGR   @0(SP),(SP)      ; Get delay loop count itself.
      SOBGR   (SP),UBLBL            ; Delay loop to slow bit tests down
      SOBGR   R1,IMBEDLBL           ; to allow Unibus DMA to occur while
      CLRL    R0                    ; testing a device register.
      .IF     NOT_BLANK, DONELBL    ; Decrement interval count
      DONELBL: SOBGR   R0,IMBEDLBL   ; Count expired, return failure
      .ENDC
      TSTL    (SP)+                 ; Pop delay loop index off stack.
      .ENDM
      .END

```


0036 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BOOTS
MAP

AUTOCONF
LIS

WRITEBOOT
MAP

BOOTSTATB
LIS

UMB
MAP

SYSGEN
MAP

ACTIMAGE
LIS

BOOT58
LIS

BOOTDEF
30L